

## Mission 1 – Docker & Co

31/03/2025

Mise en place de LAMP en conteneur sur un serveur  
GNU/Linux Debian 12 utilisant Docker



# Table des matières

<b>1. Contexte de réalisation</b> .....	3
<b>2. Introduction</b> .....	3
<b>3. Installation de Docker et Docker Compose</b> .....	3
3.1 Installation de Docker .....	3
3.2 Installation de Docker Compose .....	4
<b>4. Déploiement de la plateforme LAMP</b> .....	4
4.1 Organisation des fichiers .....	4
4.2 Configuration des conteneurs .....	5
<b>5. Lancement des conteneurs</b> .....	6
<b>6. Récupération des données de MariaDB</b> .....	7
6.1 Connexion MariaDB en ligne de commande .....	7
6.2 Page PHP pour récupérer les données .....	8

# 1. Contexte de réalisation

L'objectif de cette mission est de mettre en place une plateforme LAMP en conteneur sur un serveur GNU/Linux Debian 12 utilisant Docker. Cette installation est réalisée sur une machine virtuelle Debian 12 hébergée sur vSphere (172.18.152.40).

Le serveur est intégré dans le domaine b1XY.local et a pour adresse IP 172.18.154.142/21 avec une passerelle 172.18.152.252.

## 2. Introduction

Ce compte-rendu présente le déploiement d'une plateforme **LAMP** (Linux, Apache, MariaDB, PHP) dans un environnement conteneurisé à l'aide de **Docker** et **Docker Compose** sur un serveur **Debian**.

L'objectif est de :

- Installer et configurer Docker et Docker Compose.
- Déployer un serveur web Apache avec PHP et MariaDB.
- Vérifier le bon fonctionnement de la plateforme via une page PHP phpinfo() et la récupération de données de la base MariaDB.

## 3. Installation de Docker et Docker Compose

### 3.1 Installation de Docker

#### 1. Mise à jour des paquets :

```
root@deb41:~# apt update
```

#### 2. Installation des dépendances :

```
root@deb41:~# apt install apt-transport-https ca-certificates curl gnupg2 software-properties-common
```

#### 3. Ajout de la clé GPG et du dépôt officiel Docker :

```
root@deb41:~# curl -fsSL https://download.docker.com/linux/debian/gpg | apt-key add -  
add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian $(lsb_release -cs) stable"
```

#### 4. Installation de Docker :

```
root@deb41:~# apt update  
apt install docker-ce
```

## 5. Vérification de l'installation :

```
root@deb41:~# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-03-19 17:06:36 +04; 10min ago
 TriggeredBy: ● docker.socket
   Docs: https://docs.docker.com
  Main PID: 921 (dockerd)
   Tasks: 10
  Memory: 101.8M
   CPU: 1.075s
  CGroup: /system.slice/docker.service
          └─921 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

Docker est bien actif et en cours d'exécution.

## 3.2 Installation de Docker Compose

### 1. Téléchargement de Docker Compose :

```
root@deb41:~# curl -L "https://github.com/docker/compose/releases/download/1.26.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left    Speed
  0     0    0     0    0     0  --:--:--  0:00:01 --:--:--   0
100 11.6M 100 11.6M    0     0 3483k    0  0:00:03  0:00:03 --:--:-- 6925k
```

### 2. Ajout des permissions d'exécution :

```
root@deb41:~# chmod +x /usr/local/bin/docker-compose
```

### 3. Vérification de l'installation :

```
root@deb41:~# docker-compose --version
docker-compose version 1.26.0, build d4451659
```

Docker Compose est installé avec succès.

## 4. Déploiement de la plateforme LAMP

### 4.1 Organisation des fichiers

```
/opt/lamp-server/
├─ php-apache/
│  └─ Dockerfile
│  └─ index.php
├─ mariadb/
│  └─ Dockerfile
│  └─ init.sql
└─ docker-compose.yml
```

## 4.2 Configuration des conteneurs

### Conteneur Apache-PHP

- Création du fichier « Dockerfile » :

```
root@deb41:~# FROM php:7.4-apache
RUN docker-php-ext-install pdo pdo_mysql mysqli
```

- Ajout de la page « index.php » pour « phpinfo() » :

```
sio@deb41: ~
GNU nano 7.2 index.php *
<?php
phpinfo();
?>
```

### Conteneur MariaDB

- Création du fichier Dockerfile :

```
FROM mariadb:10.5
ENV MYSQL_ROOT_PASSWORD nopass
ENV MYSQL_USER linus
ENV MYSQL_PASSWORD nopass
ENV MYSQL_DATABASE testdb
COPY ./init.sql /docker-entrypoint-initdb.d/
```

- Création du fichier init.sql (avec des données de test) :

```
sio@deb41: ~
GNU nano 7.2 init.sql
CREATE TABLE test (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100) NOT NULL,
  email VARCHAR(100) NOT NULL
);

INSERT INTO test (name, email) VALUES ('John Doe', 'john@example.com');
INSERT INTO test (name, email) VALUES ('Jane Smith', 'jane@example.com');
```

## Fichier docker-compose.yml

```
sio@deb41: ~
GNU nano 7.2 docker-compose.yml
version: '3'

services:
  php-apache:
    build: ./php-apache
    container_name: php-apache
    ports:
      - "8080:80"
    volumes:
      - ./php-apache:/var/www/html
    depends_on:
      - mariadb

  mariadb:
    build: ./mariadb
    container_name: mariadb
    environment:
      MYSQL_ROOT_PASSWORD: nopass
      MYSQL_USER: linus
      MYSQL_PASSWORD: nopass
      MYSQL_DATABASE: testdb
    volumes:
      - mariadb_data:/var/lib/mysql

volumes:
  mariadb_data:
```

## 5. Lancement des conteneurs

1. Construction et démarrage des services :

```
root@deb41:/opt/lamp-server# docker-compose up -d
Starting mariadb ... done
Starting php-apache ... done
root@deb41:/opt/lamp-server#
```

2. Vérification des conteneurs :

```
root@deb41:/opt/lamp-server# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
e34c4aeb84b4   lamp-server_php-apache              "docker-php-entrypoi..." 2 days ago    Up 47 seconds 0.0.0.0:8080->80/tcp, [
:]:8080->80/tcp   php-apache
af5e9776fd9a   lamp-server_mariadb                 "docker-entrypoint.s..." 2 days ago    Up 48 seconds 3306/tcp
mariadb
```

Les conteneurs **php-apache** et **mariadb** sont actifs.

3. Accès à phpinfo() : Ouvrir <http://172.18.154.142:8080/index.php> dans un navigateur.

172.18.154.142:8080/index.php

**PHP Version 7.4.33** 

System	Linux 9fba51fc65c 6.1.0-32-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.129-1 (2025-03-06) x86_64
Build Date	Nov 15 2022 06:03:12
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--with-pic' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-password-argon2' '--with-sodium=shared' '--with-pdo-sqlite=/usr' '--with-sqlite3=/usr' '--with-curl' '--with-iconv' '--with-openssl' '--with-readline' '--with-zlib' '--disable-phpdbg' '--with-pear' '--with-libdir=lib/x86_64-linux-gnu' '--disable-cgi' '--with-apxs2' 'build_alias=x86_64-linux-gnu'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-mysqli.ini, /usr/local/etc/php/conf.d/docker-php-ext-pdo_mysqli.ini, /usr/local/etc/php/conf.d/docker-php-ext-sodium.ini
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902.NTS
PHP Extension Build	API20190902.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

This program makes use of the Zend Scripting Language Engine:  
Zend Engine v3.4.0, Copyright (c) Zend Technologies



## 6. Récupération des données de MariaDB

### 6.1 Connexion MariaDB en ligne de commande

```
root@deb41:/opt/lamp-server# docker exec -it mariadb mysql -uolinux -pnoPASS -e "
SHOW DATABASES;"
+-----+
| Database |
+-----+
| information_schema |
| testdb |
+-----+
```

La base « **testdb** » est bien présente.

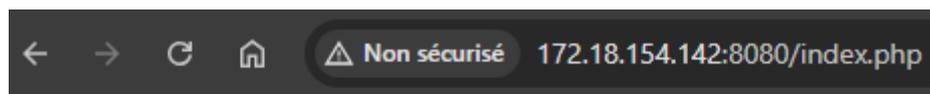
## 6.2 Page PHP pour récupérer les données

Modification de « index.php » :

```
sio@deb41: ~
GNU nano 7.2 index.php
<html>
<head>
<title>Récupération des données depuis MariaDB</title>
</head>
<body>
<table border="1">
<tr><th>Name</th><th>Email</th></tr>
<?php
$dbhost = "mariadb";
$dbuser = "linus";
$dbpass = "nopass";
$db = "testdb";
$dbconn = mysqli_connect($dbhost, $dbuser, $dbpass, $db);
if (!$dbconn) { die('Could not connect: ' . mysqli_error($dbconn)); }
$query = mysqli_query($dbconn, "SELECT * FROM test") or die(mysqli_error($dbconn));
while ($row = mysqli_fetch_array($query)) {
    echo "<tr><td>{$row['name']}</td><td>{$row['email']}</td></tr>";
}
mysqli_close($dbconn);
?>
</table>
</body>
</html>
```

Test dans le navigateur :

<http://172.18.154.142:8080/index.php>



Name	Email
John Doe	john@example.com
Jane Smith	jane@example.com

Les entrées **John Doe** et **Jane Smith** apparaissent bien.